Auditor: Dorian Hryniewicki
File: sampleAudit.sol

CVE

SWC-127 The caller can redirect execution to arbitrary bytecode locations.

HIGH
It is possible to redirect the control flow to arbitrary locations in the code. This may allow an attacker to bypass security controls or manipulate the business logic of the smart contract. Avoid using low-level-operations and assembly to prevent this issue.

Affected lines:

- #1329 - #1337

Reccomendations:
The low level call in question is ecrecover on line #1336. Consider using an alternative pattern such as the one below in the link.

- https://soliditydeveloper.com/ecrecover

SWC-123 Requirement violation.

LOW
A requirement was violated in a nested call and the call was reverted as a result. Make sure valid inputs are provided to the nested call (for instance, via passed arguments)

Affected lines:

- #1411

Notes:
Both Slither and Mythx caught a variation of this, but it seems to be a bug related to automated auditing on try/catch blocks. Exact issue can be found here:
https://github.com/crytic/slither/issues/511

SWC-103 A floating pragma is set.

LOW

It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Affected lines:

- #5
- #15
- #25
- #37
- #51
- #64
- #130
- #176
- #254
- #281
- #366
- #591
- #621
- #649
- #676
- #707
- #852
- #881
- #1471
- #1533
- #1617

Recommendation:

Set pragma to a specific compiler to avoid unforeseen exploits.

Additional Findings

Hard-coded values

LOW

Some state-variables are initialized with values.

Affected lines:

- #19

- #1664
- #1666
- #1667

Recommendation:
Consider setting default values within the constructor with the relative setter functions. As an alternative, you can also have a conditional fallback value within the relative getter functions.

Missing events access control

LOW
Missing events for critical access control parameters.

Affected lines:

- #1570
- #1701
- #1711
- #1716
- #1749
- #1814

Recommendation:
Emit events on critical state changes to track activity off-chain.

Low-level calls

INFORMATIONAL
The use of low-level calls is error-prone. Low-level calls do not check for code existence or call success.

Affected lines:

- #425
- #499
- #526
- #553

Recommendation:
Avoid low-level calls. Check the call success. If the call is meant for a contract, check for code existence.

Unused function parameters

INFORMATIONAL
Parameters that are never used by their functions exist.

Affected lines:

- #1689
- #1694

Recommendation:
Consider removing these from function signatures.

Missing Licensing

INFORMATIONAL
SPDX license identifier not provided in source file.

Recommendation:
Before publishing, consider adding a comment containing "SPDX-License-Identifier: <SPDX-License>" to each source file. Use "SPDX-License-Identifier: UNLICENSED" for non-open-source code.

Assembly Usage

INFORMATIONAL
The use of assembly is error-prone and should be avoided.

Affected lines:

- #575-#578
- #1350 - #1354
- #1418 - #1420
- #1639 - #1643